

Guide Pratique

Deployer Ghost CMS avec Docker

Installation complete en 15 minutes — auto-heberge

Mars 2026

Sommaire

1. Prerequis
2. Installation de Docker
3. Structure des dossiers
4. docker-compose.yml
5. config.production.json
6. Demarrage et verification
7. Configuration DNS Cloudflare
8. Reverse Proxy SSL (Zoraxy / Nginx)
9. Commandes de maintenance

1. Prerequis

Avant de commencer, assurez-vous que votre serveur dispose de :

- Ubuntu 20.04 LTS ou superieur (Debian/Red Hat compatible)
- Acces root ou sudo sur la machine cible
- Docker Engine 24+ (voir Section 2 pour l'installation)
- Docker Compose v2 (plugin docker-compose-plugin)
- Un nom de domaine pointant vers l'IP du serveur
- Ports 80 et 443 ouverts dans le pare-feu
- Un compte Cloudflare ou autre DNS pour le certificat SSL

2. Installation de Docker

Si Docker n'est pas encore installe sur votre serveur Ubuntu :

```
# Mettre a jour les paquets
sudo apt update && sudo apt upgrade -y
sudo apt install -y ca-certificates curl gnupg

# Ajouter la cle GPG officielle Docker
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
    | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# Ajouter le depot
echo "deb [arch=$(dpkg --print-architecture) \
    signed-by=/etc/apt/keyrings/docker.gpg] \
    https://download.docker.com/linux/ubuntu \
    $(. /etc/os-release && echo $VERSION_CODENAME) stable" \
    | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io \
    docker-buildx-plugin docker-compose-plugin

docker --version && docker compose version
```

i Documentation officielle : <https://docs.docker.com/engine/install/ubuntu/>

3. Structure des dossiers

```
sudo mkdir -p /mnt/docker-data/ghost-blog/content

# Arborescence :
# /mnt/docker-data/ghost-blog/
#   +-- content/          <- donnees Ghost (images, themes, DB)
#   +-- docker-compose.yml
#   +-- config.production.json
```

4. docker-compose.yml

```
version: '3.8'
services:
  ghost:
    image: ghost:5
    container_name: ghost-blog
    restart: unless-stopped
    ports:
      - "2368:2368"
    environment:
      NODE_ENV: production
    volumes:
      - ./content:/var/lib/ghost/content
      - ./config.production.json:/var/lib/ghost/config.production.json:ro
```

5. config.production.json

```
{
  "url": "https://blog.votredomaine.com",
  "server": { "port": 2368, "host": "0.0.0.0" },
  "database": {
    "client": "sqlite3",
    "connection": {
      "filename": "/var/lib/ghost/content/data/ghost.db"
    }
  },
  "mail": { "transport": "Direct" },
  "logging": { "transports": ["stdout"] },
  "process": "local",
  "env": "production"
}
```

i Remplacez `blog.votredomaine.com` par votre domaine reel.

6. Demarrage et verification

```
cd /mnt/docker-data/ghost-blog
docker compose up -d

docker ps | grep ghost-blog
docker logs -f ghost-blog

# Test local (doit retourner 200)
curl -s -o /dev/null -w '%{http_code}' http://localhost:2368
```

7. Configuration DNS Cloudflare

1. Connectez-vous a dash.cloudflare.com et selectionnez votre domaine
2. DNS > Records > Add record

3. Type : A | Name : blog | IPv4 : votre-IP-serveur
4. Proxy status : Proxied (nuage orange) | TTL : Auto | Save
5. Propagation DNS : 1 a 5 minutes avec Cloudflare

8. Reverse Proxy SSL

Option A — Zoraxy (interface web)

1. Accédez a votre interface Zoraxy
2. Nouvelle regle : blog.votredomaine.com -> localhost:2368
3. Activez HTTPS avec Let's Encrypt integre

Option B — Nginx

```
server {  
    listen 443 ssl;  
    server_name blog.votredomaine.com;  
    ssl_certificate      /etc/letsencrypt/.../fullchain.pem;  
    ssl_certificate_key  /etc/letsencrypt/.../privkey.pem;  
    location / {  
        proxy_pass http://localhost:2368;  
        proxy_set_header Host $host;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
}
```

9. Commandes de maintenance

```
docker restart ghost-blog  
docker logs --tail 50 ghost-blog  
docker pull ghost:5 && docker compose up -d  
  
# Sauvegarder la base de donnees  
cp /mnt/docker-data/ghost-blog/content/data/ghost.db \  
~/backups/ghost-$(date +%Y%m%d).db  
  
docker compose down
```

Article complet : blog.botum.ca/deployer-ghost-cms-docker

Site web : www.botum.ca • contact@botum.ca •

— Canada